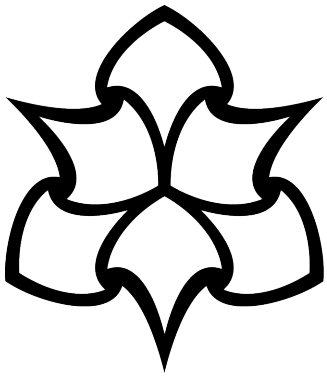

ORIGINAL ARTICLE

Journal Section

A Novel Dynamic Framework to detect DDoS in SDN using Meta Heuristic Clustering

Muhammad Shakil[1] | Dr. Alaelddin Fuad Yousif Mohammed[1] | Rajakumar Arul[2] | Dr. Ali Kashif Bashir[3] | Dr. Jun Kyun Choi[1]

[1]Korea Advanced Institute of Science and Technology (KAIST), South Korea

[2]Department of Computer Science Engineering Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru

[3]School of Computing, Mathematics, and Digital Technology, Manchester Metropolitan University, United Kingdom.

Correspondence
Rajakumar Arul, Department of Computer Science Engineering,l Amrita School of Engineering, Bengaluru, India
Email: rajakumararul@ieee.org

Funding information
This work was supported by the KAIST GCORE (Global Center for Open Research with Enterprise) grant funded by the Ministry of Science and ICT (Project SDN/NFV and Cloud).

Security is a crucial factor in the continuously evolving programmable networks. With the emergence of programmable networking terminals, the need to protect the networks has become mandatory. Software Defined Network (SDN) provides programmable switches thereby isolating the data plane from the control plane. Many security algorithms have been proposed to protect the network, however, they have failed to protect SDN from attacks like Distributed Denial of Service (DDoS), Jamming, and Man-in-the-Middle. In this article, we only address DDoS attack that prevails in the SDN networks. Isolation of control plane from the data plane increases the probability of an attack on the data plane. Therefore, a framework that can handle the dynamic traffic and can protect the network from DDoS attacks is required. Our proposed Whale Optimization Algorithm Based Clustering for DDoS Detection (WOA-DD) avoids the DDoS attacks using a meta-heuristic approach by clustering the attack requests. We evaluated this algorithm for robustness in comparison with several existing solutions and found it to be safe under several conditions. The proposed attack request clustering is explored to check its feasibility with various

machine learning approaches and found to be stable with the prevailing mechanisms. Analysis of the algorithm under varied conditions reveals that WOA-DD is robust, stable and efficient against DDoS attacks.

KEYWORDS

SDN, Security, DDoS, Clustering, Optimization, Cryptography

1 | INTRODUCTION

The Software Defined Network (SDN) is an advantage for the networking field as it satisfies the growing demands of companies and data centers. SDN facilitates the role of network administrators, where they can administer the network through a code that abstracts the functionality of the lower level layers. SDN includes the data plane and the control plane, where the data plane includes packet messages generated by network devices and the centralized and programmable control plane is responsible for routing the message to the intended recipient. As with the increasing commercial demands, the traditional network architecture suffers from the problem of initial configuration, installation and reconfiguration, which requires a remarkable experience. In addition, removing or moving a device requires more costs and also the process requires a lot of time. Most network devices are dependent on the provider, which serves as an obstacle to network applications and is inflexible in handling various business demands that increase day by day.

Traditional static network architecture finds it difficult to manage scalable and dynamic business environments, as well as the storage required by emerging technologies such as virtualization, large data centers and clusters, which paved the way for a more dynamic architecture called Software Defined Networking. Traditionally, the data plane and the control plane are grouped into a network device. The control plane is in charge of giving the essential data for creating a forwarding table. The control plane uses several shorter path algorithms, such as the shortest Open source path, the Floyd's algorithm and the Dijkstra's algorithm, etc. to find the path between any network devices. The data plane knows which packets to send but does not know how to send them, so it uses data from the forwarding table. Using the forwarding table, the network device sends the packet it received to the desired recipient. In SDN, the control plane function is summarized and placed in the SDN controller. An SDN controller is a piece of code running on a server responsible for communicating with data planes from various network devices and continuing packets efficiently while also managing network traffic significantly. Logically, SDN owns a centralized architecture, where SDN controllers in remote locations handle the flow demand of various network devices.

Although SDN is a blessing, there are some weaknesses due to its centralized nature[1]. Firstly, the control plane and the data plane are separated when the network device receives the packet. The data plane in a network device that does not know the flow table sends requests to controllers in remote locations that cause an overhead in the message passing. Secondly, as new network devices are transpiring exponentially, the controller has to handle a greater number of flow requests without affecting the response time. Lastly, the distance between the controller and the network device is responsible for latency. As the distance between the controller and the switch increases, communication delay among them also increases, which in turn increases latency. Distributed SDN controllers are used to address the issues

mentioned above. SDN differs from the traditional network architecture which vertically integrates the data plane and control plane in the networking device. Additionally, middle box functions like a firewall that is used to handle the traffic. Though SDN makes the network management easier, it suffers from a serious drawback of resilience. Since the controller is decoupled, if it fails or is unavailable, the flow request made by the networking device will not be handled efficiently. While designing SDN it is essential to check that even if a node fails, the computation should be carried on as if the node was present[2].

When the network grows exponentially, security is another prime challenge to address [3]. Blockchain technique is a new buzzword integrated with crypt arithmetic algorithm that provides security in a trust-less manner without any intermediary [4, 5]. Each SDN controller is viewed as a block. Each block has the data of the controller, the hash value of the data and hash value of the previous block. A blockchain is viewed as a distributed ledger that is visible to anyone in the network. Once some data is entered in a block, the data cannot be modified i.e. it is immutable. When someone tries to modify the data in a block, the new hash value of the data will be generated which is different from its successor block thereby invalidating the block. Thus, the blockchain offers a high level of security. The SDN is effective in network management, thus overriding traditional protocols like SNMP. The principle that SDN offers is a high level of security which lies in the fact of decoupling of data plane and control plane. This provides effective security policy which can be customized then and there [6, 7]]. The centralized SDN controller should aim to offer a high level of security with minimal response time for routing the packets in a high-end network. When the controller receives the flow request from any networking device, it responds with flow rule such that the data plane in networking device routes the packet to the intended recipient efficiently. Intrusion Detection System (IDS) and Traffic Analyzer tools placed in the network generates security related data which is frequently transferred to the controller. Since the controller is a piece of code, it can be easily programmed to include functions to correlate the security data from IDS with the behavior of the network. Thus, it upgrades the security policy in the flow table and then forwards the flow rule to the networking device[8]. Though the features of SDN, such as centralization and programmability, are attracting the enterprises for their provisioning of high level security, these can be easily compromised. The centralized SDN controller is highly vulnerable to cryptographic attacks, which leads to single point failure thereby failing to provide confidentiality, integrity and authentication.

The most serious threat that continuously floods the network with requests from the adversary in order to waste or exhaust network resources such as bandwidth and memory leads to increase in network traffic which is called Denial of Service (DoS)[9]. The power of DoS attacks increases exponentially when the attack is executed by multiple adversaries which are termed as Distributed Denial of Service (DDoS). In order to handle DDoS attacks, the controller deploys some additional applications which effectively prevent the target networking device from malicious users. Since the controller is a software, it can be programmed to block DDoS attack by enhancing the security strategy. This results in dropping malicious packets which help in reducing the network traffic. Consequently, it protects the target [10]. Though SDN offers a high level of security by enforcing the security policy, the question of how to protect the control plane from DDoS is a serious challenge which makes the researchers exploit more techniques to tackle a security breach where an adversary can travel [11]. By depleting the network resource, the centralized controller is unavailable for legitimate users i.e. the entire network is compromised. Thus, it contributes in denying the requests made by non-adversary networking devices. Whenever a networking device receives a new packet, it checks for the packet's match in its flow table. If the match is not found, it forwards the request to the controller. The malicious packet will have a spoofed IP address as it does not match with any entry in the flow table. The rate at which the controller receives the packet increases exponentially, enabling malicious users exhaust the resources of the controller, hence making the controller unavailable for legitimate users [12].

In recent days the machine learning approaches, such as unsupervised learning, aim to create a model for grouping the normal requests with attacks. The optimal detection of attacks is a serious concern. Thus, in this paper, Meta Heuristic algorithm based clustering is used to provide an optimal solution for approximate problems. Whale Optimization algorithm [7, 13]] mimics the behavior of whales that is used in the SDN in order to detect attacks before they happen, thereby maximizing the accuracy and protecting the entire network from crash.

The rest of the paper is organized as follows. Section II presents the state-of-the-art works, Section III describes the system model of the SDN with a frame work named Whale Optimization Algorithm based DDoS Detection (WOA-DD) algorithm using meta heuristic approach and Section IV details the mathematical analysis of the system under discrete and continuous traffic. Section V details the implementation and the analysis of WOA-DD algorithm. In Section VI concludes the paper with possible future works.

2 | RELATED WORK

Software Defined Networking (SDN) is used to protect the network from Distributed Denial of Service (DDoS) attacks [14] Cloud computing provides services for optimal provisioning of resources which manages computation and storage effectively. SDN facilitates effective network management. The integration of cloud computing and SDN complicates the network security. Some threats to security are Denial of Services (DoS) and Distributed Denial of Services which make the services unavailable for legitimate users by exhausting the network resources. Conventionally, network security is provided via network administrator by placing a piece of hardware in order to handle DDoS attacks, whereas in SDN it is not possible as everything is programmable. Thus, when SDN is designed properly it can effectively handle the DDoS attacks. DaMask (DDoS attack mitigation architecture using software defined networking) is proposed to address the security challenges brought by cloud computing. The framework includes two phases DaMask-D which is to detect any DDoS attacks designed using anomaly detection mechanism and DaMask-M which is to mitigate the attack. In addition, DaMask-D phase addresses the data-set drift problem which is a variation in the network traffic conditions that occurs while building and using the model. The acronyms in this sections are tabulated in Table 1.

The DDoS attacks are either resource focused attacks that exhaust the network bandwidth and memory etc. or application-oriented attacks which depletes the web services. The attackers in the cloud may reside in private network, public network or both. The challenges introduced by cloud computing are called extended defense perimeters and rapid resource allocations. These challenges refer to the public accessibility of resources. This could also be referred to as network topology. SDN decouples data plane and control plane of network elements by placing the control plane in the centralized network controller. The other feature which attracts SDN is network virtualization which hides network topology from the control program. Though, SDN offers benefits for invading DDoS attack due to communication between data plane and control plane, the centralized controller and the traffic overhead may serve as a victim for new attacks. SDN should be designed with care for easy packet transferring and to avoid single point failure. Detection and blocking of DDoS attacks using anomalous traffic strategy are difficult and inefficient [15]. The alternative for handling DDoS is the use of SDN in order to enhance security.

The SDN-based DDoS blocking scheme is designed to handle DDoS where the application is running in the centralized SDN controller. Whenever the server in SDN network identifies the existence of DDoS, it communicates it with the blocking application running in SDN controller which in turn provides a redirected IP address to access the

TABLE 1 List of Acronyms

Abbreviation	Explanation
SDN	Software Defined Networking
DDoS	Distributed Denial of Service
WOA-DD	Whale Optimization Algorithm Based Clustering for DDoS Detection
SNMP	Simple Network Management Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
ALTO	Application Layer Traffic Optimization
MIB	Management Information Base
CDN	Content Delivery Network
API	Application Programming Interface
SPRT	Sequential Probability Ratio Test
TCAM	Ternary Content -Addressable Memory
QOS	Ouality of Service
C-WOA	Chaotic Whale Optimization Algorithm
DaMask	DDoS attack mitigation architecture using SDN network
T-Table	Temple Table height

server. Further, the application broadcasts the redirected IP address to the legitimate network elements in order for them to access the service. The problem with this approach is that there is a need for co-operation between the SDN controller and the server to be protected, which is difficult in practice. In a dedicated firewall or DDoS, an appliance is needed to protect layer 2-7 from behavioral attacks [16]. Flow is defined as the sequence (a defined order) of packets that should be maintained. The flow can be characterized into four classes viz. short-lived small flow, short-lived large flow, long-lived small flow and long-lived large flow. A flow which exceeds minimum bandwidth and minimum observation interval is considered as long-lived large flow. Some examples of DDoS attacks are SYN flood attacks, UDP flood attacks, Christmas tree flood attacks and Tenant misbehavior. If a DDoS attack produced by long-lived large flow is detected, various types of actions such as rate-limiting, re-marking and discarding are done on the service flows based on the configuration. This is needed to prevent the exhaustion of resources. Flow-aware sampling provides the substantial benefit of reducing the number of samples needed for DDoS detection as long-lived large flow is automatically detected in edge switch/ router or border router.

In SDN based content delivery, the network is protected against DDoS using multi-defense strategy [17]. As data and control plane are decoupled in SDN, the data plane is integrated with SDN applications which co-ordinate with Application Layer Traffic Optimization (ALTO) servers. Each CDN is monitored through ALTO server which provides network and cost map controlled by SDN controller including Management Information Base (MIB). The controller can adjust its open flow table if any traffic goes beyond the acceptance level using MIB. When the controller includes rules in the open flow table, it sends them to the data plane present in switches. In addition to network and cost map, the ALTO server is integrated with marking path map which contains the path taken by the request packet. The CDN together

with ALTO is protected using firewall against intruders. The CDN servers are secured using protection switches which utilize marking path maps and MIB to decide whether the traffic is a DDoS attack or is it legitimate.

Though the mechanism of marking path map and MIB helps to invade against DDoS and the bot it comes from, it fails to find the real launcher. SDN [7] is used to balance the load during DDoS attacks. Two level load balancing algorithm aids in balancing the load between servers and network devices. The load balancing mechanism uses only source and destination IP address. The algorithm aims to maximize the survival time of the system under DDoS attack and thereby reducing the severity of such behavioral attacks. The proposed solution of DDoS mitigation for SDN network is divided into active techniques. These techniques not only filter traffic deemed as an attack but also provide survival techniques which enhance the survival time of the network elements in addition to performing effective load balancing. SDN controller chooses an alternative path between source and destination as the current path is overloaded, thereby distributing network traffic. L7 and L4 load balancing mechanisms are used to split the traffic between endpoint servers and the packets across different paths in the network. Though, the system provides DDoS mitigation solution to maximize the survival time of the system under attack, it is prone to single point failure which degrades the performance of the entire network. Single point failure in SDN network by flooding the controller through malicious user was addressed [18]. If the controller is under DDoS attack, the network bandwidth between the controller and the switch is occupied by a malicious user which degrades the performance of the network.

The proposed mechanism uses two crucial parameters to decide whether the traffic is normal or is an attack viz. the minimum number of packets per connection and the average number of connections of frequent users within a stipulated time interval. The proposed method to handle DDoS uses temple table (T- table) in the controller. The T-table contains a source IP address and a counter which keeps track of the number of arrived packets from the source. Whenever the controller receives a request from the switch, it increments the counter value by 1. When the counter value is equal to the average number of connections, the number of packets transferred will be compared with the minimum number of packets per connection. If the number of packets transferred is greater, the traffic is legitimate else the traffic is termed as an attack. Although this method reduces the severity of the DDoS, it is not suitable for huge traffic. Single point failure [19] of the controller in SDN by DDoS is handled using an entropy variation of destination IP address. The centralized controller is a significant advantage of SDN since it is easily programmable and is a boon to today's network. The issue becomes worse when the controller is made unreachable by attackers that affect the quality of service.

Early detection of DDoS in the controller is achieved by measuring randomness of incoming packets. Randomness is measured through entropy. The DDoS detection mechanism in controller uses two components viz. window size and threshold. Entropy is calculated for packets within the window. If the entropy is below the threshold, then the attack is detected. The controller monitors the destination IP address and the number of times it is requested in a hash table. When the window has all IP address as unique, the entropy is maximum, else it is minimum. In other words, when an attack is directed towards a particular host, the window has most of the packets with intended host's IP address which, as a result, brings entropy down. This methodology detects the DDoS attack efficiently but does not mitigate it. Exploitation of SDN controller by DDoS is detected and mitigated using attack time pattern [20]. Whenever a new packet arrives at the switch, it checks its own flow table. If there exists a match, the packet is forwarded to the intended destination specified in the flow, else it forwards the request to the controller. The controller then checks its flow table for a match. If a match is found, the flow rule will be sent to all the networking devices, else the packet is dropped. The adversary utilizes the size of the flow table in the controller and the switch which then sends a huge amount of spoofed IP addresses. If the address is new, the switches forward it to the controller. In case if the arrival rate of the packets

at the controller is high, the resources of the controller are exhausted and it then stagnates the legitimate traffic. To detect a DDoS attack, the collector is deployed in the SDN controller. When the controller receives a packet with a new destination that is not in its flow table, it forwards the packet to the flow collector which marks it invalid. When an invalid packet reaches a threshold, it sends a notification to the controller. The controller then creates a new flow rule for the invalid packets to be sent directly to the flow collector and forward it to all devices.

Mitigation based on clustering attack time pattern is not previously discussed. SDN serves as a great tool to defeat DDoS in cloud computing [21]. In the meantime, SDN itself is vulnerable to DDoS. The DDoS attacks can occur at application, control or infrastructure layer of SDN. The application layer DDoS attack either occurs at applications or northbound Application Program Interface (API). The control layer DDoS attack is crucial and leads to single point failure which either degrades or shuts down the entire network services. The attack is made by attacking the controller, the northbound API, southbound API, westbound API or eastbound API. By sending a huge amount of spoofed IP packets, the bandwidth and flow table of the controller is overwhelmed, thereby making the controller unreachable. The infrastructure layer of the DDoS attack violates the data plane in the network elements. Two ways to launch this attack are attacking some switches and attacking southbound API. By sending a large amount of spoofed packets, the header is sent to the controller for flow rule and the packet has to reside in the flow table of switches, thus saturating the memory of networking devices. A collaborative approach is used to detect DDoS in SDN network [22].
A monitor placed in the network is used to observe the network traffic. The correlator present in the network switch examines the packet thoroughly for any anomalies. SDN controller modifies the flows in order to reduce the severity of the attack. When the clients send the request to the server, the information will also be sent to the monitor. The monitor is integrated with Intrusion Detection System. If the monitor detects an attack, it sets the alert flag and then alerts the controller. The correlator in controller requests the flow table from the switch. If the address is spoofed then it is detected as malicious. The collaborative approach involving alerting, detection and mitigation takes less time to resolve the DDoS attack.

Sequential probability ratio test (SPRT) is proposed to detect a DDoS attack that shuts down the entire network by targeting controller [13]. A DDoS attack is difficult to identify since switches and controllers can't identify the malicious flow. Further, DDoS is a reflection-based flooding attack where the attacker generates the flow using TCP, UDP and ICMP etc. The SPRT analyzes the traffic flow as normal or low traffic to check whether the controller is under attack. It made an assumption that each switch is capable of obtaining statistic information of arriving flows and sends the report to the controller. The SPRT outperforms percentage based detection, count-based detection and entropy-based detection in terms of accuracy. DDoS attack detection in SDN involves victim-detection and post-detection [23]. Victims can be detected by monitoring flow volume and flow rate asymmetry. The size of memory plays a prominent role in the detection of any behavioral attack. The memory used in SDN switches is Ternary Content- Addressable Memory (TCAM) which is very expensive and power consuming. Since the memory of the switches is limited, victim-detection is done through minimizing the maximum granularity of all ranges of IP addresses.

The post-detection procedure can be passive or active. The passive detection involves asking the victim to provide service at a new IP address. But this detection mechanism is not advisable as it is time consuming and wastes the computational resources. Having found the attacker's IP address, the active detection procedure creates a flow in the controller and broadcasts it to all the network switches. If the TCAM size is limited then the sequential method is advisable to find the victim of the attack. On the other hand, if the TCAM size is greater, then the concurrent method is used to find the attacker. A framework for detecting and mitigating DDoS using sFlow and Open-Flow is implemented [24]. This framework works with the constraint that there can be only a single application session for an HTTP request

from a single IP address. The Open-Flow controller is used to check whether the incoming traffic flow is normal or from a malicious user. When the traffic is found as abnormal, rather than creating a new flow rule, the framework uses sFlow and flow statistics collector to perform mitigation. Experimentation results reveal that the framework outperforms QoS based approach. Drawbridge based on traffic engineering is used to handle a DDoS attack [25]. This framework blocks unwanted traffic while still allowing normal traffic. The controller and hosts must subscribe to the traffic engineering service provided by Internet Service Provider. Whenever a new flow arrives that is missing from the flow table of the controller before deploying the new flow, it verifies the rules and determines whether or not to deploy those rules in the switches. The framework is scalable and can tackle situations where numerous victims are under threat. SDN replaces the traditional networking and serves as the best platform for DDoS detection and mitigation [26]. The decoupling of data and control plane serves as a root for DDoS attacks as the attacker may launch the attack by creating too many flows. DDoS attacks are of two classes. The first class of attacks is resolved by balancing the traffic in SDN while the second class of attacks is resolved by keeping track of the source of the traffic.

3 | SYSTEM MODEL

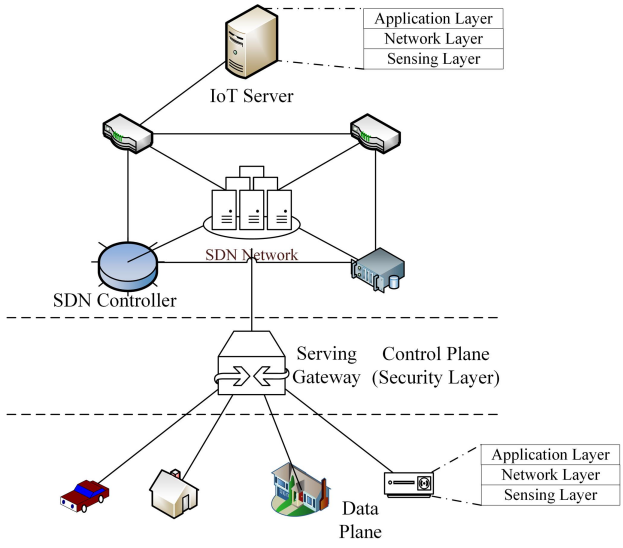

Figure 1. DDoS attack Scenario in SDN

SDN is used to handle Distributed Denial of Service (DDoS) attacks but the problem with SDN is that when the controller is flooded with attacks, the entire network will be compromised. The DDoS aims to attack the SDN controller by overflowing the flow table in the data plane as shown in figure 1. Due to the limited memory and cost, the flow table in the data plane is little. Thus, whenever a request with an unknown entry in the flow table arises, switches in the data plane forward the requests to the controller. The controller checks its flow table and if the request is legitimate it responds with a valid flow. When the requests received are more in a specified time interval, the time taken for the controller to look up the flow table and respond also increases which exhausts the resources of the controller and makes

the controller unavailable for handling legitimate requests. Thus, prior to forwarding the request to data plane, it is necessary to check whether the request is an attack or not.

3.1 | Whale Optimization Algorithm Based Clustering for DDoS Detection (WOA-DD)

Whale Optimization Algorithm is a nature-inspired Meta Heuristic algorithm. The goal of WOA-DD is to group the historical set of data into clusters viz. normal and DDoS attack. After clustering, whenever a new request arrives it will be assigned to any one of the clusters and the cluster then will be reformed again. The notations used in this section is given in Table 2.

TABLE 2 List of Notations

Notation	Explanation
$\|W\|$	Number of Solutions
W_i	Represents the i_{th} solution
c_i	Represents the i_{th} cluster centroid
A_k	Represents the k_{th} of the request
Req_i	Represents the i_{th} Request
$f(W_i)$	Represents the fitness of the i_{th} whale
n_i	Represents the number of instances that fall into cluster c_i
$d\left(c_i, (Req_i)_{ci}\right)$	Represents distance between i_{th} cluster centroid and the request that fall into the corresponding cluster
$X_{Req_i c_i}$	Represents Boolean variable to indicate whether Req_i
W_{Best}	Represents the position of the best whale
$\overrightarrow{A(t)}$	Coefficient vector at the iteration t
$\overrightarrow{C(t)}$	Coefficient vector at the iteration t
l	Constant in the range (-l,l)
p	Constant in the range (0,1)
$\overrightarrow{Dis}$	Represents distance between two whales
$\overrightarrow{a}(t)$	Represents a vector initialized to 2
$\overrightarrow{r}(t)$	Represents a vector initialized to (0,1)
λ	Represents inter-arrival time
μ	Represents service time
p	Represents utilization time

3.1.1 | Representation of Whale

Let the number of whales be given as $|W|$ where W represents the population of whales signified as $W \leftarrow \{W_1, W_2, \ldots . W_m\}$. Each whale W_i represents an optimal solution. Thus, the population of whales represents the candidate solution for clustering. In the problem of detecting DDoS attacks, the whale W_i has 2 dimensions that correspond to 2 cluster centroids for a normal and a DDoS attack. The representation of whale W_i is given in equation (9):

$$W_i \leftarrow \{c_1, c_2\} \tag{9}$$

Where each cluster centroid c_i represents the request attributes of Req_i represented in equation (10):

$$c_i \leftarrow \left\{\{A_1, A_2, , , , A_D\}_{Reqi}, \{A_1, A_2, , , AD\}_{Reqj}\right\} \tag{10}$$

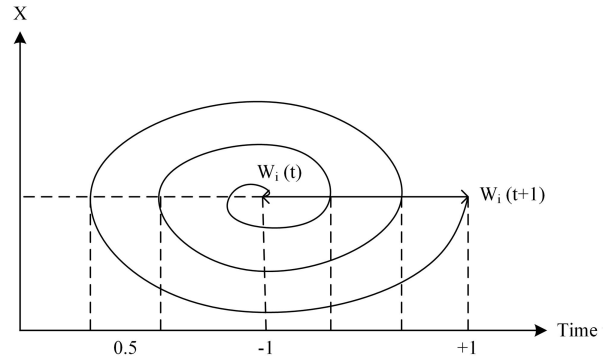

Figure 2. Whale updating position in a spiral representation

3.1.2 | Fitness

The fitness function is used to evaluate each whale. The objective of any clustering is to minimize the intra-cluster distance and maximizing the inter-cluster distance. In WOA based clustering, each request is viewed as an instance in solution space and will be represented as a point in the Euclidean distance. At each iteration, the whale adjusts its position based on the best position in the population as shown in figure 2. The fitness function is the minimization of the average distance between the cluster centroid and the request represented in equation (11):

$$Min(f(W_i)) \leftarrow \frac{\sum_{i=1}^{2} \frac{\sum_{j=1}^{n_i} d\left(c_i,(Req_i)_{ci}\right)}{n_i}}{2} \tag{11}$$

Subject to

$$\forall Req_i d\left(c_i, (Req_i)_{ci}\right) < d\left(c_k, (Req_k)_{ck}\right) where i \neq k$$

$$\forall Req_i \; Req_i \in c_i$$

$$X_{Req_i c_i} \leftarrow \begin{cases} 1 & \text{if } Req_i \in c_i \\ 0 & \text{else} \end{cases}$$

The proposed WOA-DD takes a dataset and cluster the instances in the dataset into two clusters viz. Attack and normal requests, based on the attributes. When a new request arrives, the algorithm predicts whether it is attack or normal, by computing the distance with the two cluster centroid computed. When the new request arrives, it will fall into cluster which has minimum distance. By this way, each request will be verified before forwarding thereby if the request is detected as attack, it will not be forwarded to controller.

The algorithm for clustering the normal requests and DDoS attacks is represented in Algorithm 1:

Algorithm: WOA-DD

Input: $Req \leftarrow \{Req_1, Req_2, ..., Req_n\}$
Output: $c_1(Normal requests), c_2(DDoS Attacks)$
1. For t $\leftarrow$ 1 $to Max\_iteration$
2. For each whale W_i
3. Compute fitness $F(W_i)$
4. End for
5. Choose the best whale W_{best} $\leftarrow$Min(f(W_i))
6. For each whale W_i
7. Compute $\overrightarrow{A(t)}$ using equation (14)
8. Compute $\overrightarrow{C(t)}$ using equation (15)
9. Choose l randomly in the interval (-1, 1)
10. Choose p randomly in the interval (0, 1)
11. If p < 0.5
12. If $|\overrightarrow{A(t)}|$ < 1
13. Compute the Position of the Whale using equation (13)
14. Else if $|\overrightarrow{A(t)}|$ > 1
15. Select a Whale W_{rand} randomly from W
16. Compute the Position of the Whale using equation (19)
17. End if

18. Else if p ≥ 0.5
19. Compute the Position of the Whale using equation (17)
20. End if
21. End for
22. End for
23. Return W_{Best}

The WOA-DD algorithm does exploration and exploitation in order to find an optimal solution. It includes three phases i) Encircling prey ii) Bubble net attacking (Exploitation) iii) Searching for prey (Exploration). In the Encircling prey phase, each whale is assigned a position in which the request has been made. Each whale computes its own fitness and the preeminent fitness among the population is found. All other whales tend to move towards the whale W_{Best} which has the minimum fitness. The distance between the whale W_i and the whale W_{Best} is computed using equation (12):

$$\overrightarrow{Dis} \leftarrow |\vec{C} * \overrightarrow{W_{Best}(t)} - \overrightarrow{W_i(t)}| \tag{12}$$

With respect to distance, each whale W_i updates its position represented in equation (13):

$$\overrightarrow{W_i(t+1)} \leftarrow \overrightarrow{W_{Best}(t)} - \overrightarrow{A(t)}.\overrightarrow{Dis} \tag{13}$$

where t represents current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors calculated using equations (14) and (15):

$$\overrightarrow{A(t)} \leftarrow 2^{*}\overrightarrow{a(t)}^{*}\overrightarrow{r(t)} - \overrightarrow{a(t)} \tag{14}$$

$$\overrightarrow{C(t)} = 2^{*}\overrightarrow{r(t)} \tag{15}$$

$\vec{a}$ is linearly decreased from 2 to 0 and $\vec{r}$ is a random vector in the interval (0, 1).

In the exploitation phase, whales attack their prey either by using shrinking encircling mechanism or spiral updating position. Shrinking encircling mechanism works by decreasing the value of $\vec{a}$ linearly from 2 to 0, thereby decreasing the coefficient vector $\vec{A}$, as $\vec{A}$ is always represented as [-a, a]. The new position of the whale W_i is computed as a position between the original position and the position of the best whale W_{Best}. All whales move towards the best whale in a spiral path which is computed using equation (16):

$$\overrightarrow{W_i(t+1)} \leftarrow \overrightarrow{Dis'}.e^{bl}.cos(2\Pi l).\overrightarrow{W_{Best}(t)} \tag{16}$$

$$\overrightarrow{Dis'} \leftarrow |\overrightarrow{W_{Best}(t)} - \overrightarrow{W_i(t)}| \tag{17}$$

where b is a constant. Since the whales move in a shrinking circle and in a spiral path simultaneously; to model this behavior a random integer p represents the probability of switching between shrinking circle and spiral model. In the exploration phase, based on the value of the coefficient vector $\vec{A}$, the whales can randomly search for the best position.

When the absolute value of $\vec{A}$ is greater than 1, then a random whale W_{rand} is chosen from the population. The position of other whales W_i is updated based on random whale W_{rand} :

$$\overrightarrow{Dis} \leftarrow |\vec{C} * \overrightarrow{W_{rand}(t)} - \overrightarrow{W_i(t)}| \tag{18}$$

$$\overrightarrow{W_i(t+1)} \leftarrow \overrightarrow{W_{rand}(t)} - \overrightarrow{A(t)}.\overrightarrow{Dis} \tag{19}$$

Here, with respect to the SDN, each request is assumed to be a whale. Whenever a request is mapped to the corresponding cluster, then next whale will be chosen random using the equation (18) and (19). Thus, mapping the attack request to its appropriate cluster not only protects the system from attack, it also reduces the load that may be imposed on SDN controller for processing these requests thereby overall efficiency is improved through our proposed WOA-DD mechanism.

4 | MATHEMATICAL ANALYSIS

In this article, we have formulated our proposed methodology with respect to the deterministic property of the traffic in the network. Based on the network conditions, we have formulated two cases viz. continuous and discrete. Here, we call the traffic to be discrete if it is deterministic and continuous if it is non-deterministic. When the traffic is assumed to be deterministic, it follows the Poisson distribution which has been analyzed for our proposed methodology using the queuing model under Case 1 and for the un-deterministic continuous traffic the same is analyzed under Case 2.

Case 1: When the network traffic is discrete. The simplest queuing model that can be used for the discrete traffic is Poisson Model. The Poisson distribution is attracted for discrete traffic due to its memory-less state property. The inter-arrival time of the packets is given by λ per unit time and the service time is given by μ.The utilization rate is given in equation:

$$p \leftarrow \frac{\lambda}{\mu}$$

The number of requests arriving in any time interval is determined using the probability density function of Poisson process represented in equation:

$$P(\#RequestintimeIntervalT) \leftarrow \frac{e^{-\lambda T}(\lambda T)^{\#Request}}{\#Request!}$$

The utilization rate p has an attracting behavior to decide whether the requests are normal or attack. When the utilization rate p<1 , the inter arrival time of the packets is less than the service time of the packets, there will be no waiting time. On the other hand, when the utilization rate p>1the number of packets arriving in unit time exceeds the number of packets being served, that leads to the exhaustion of the network resources which as a result influences the occurrence of Denial of Service attack.

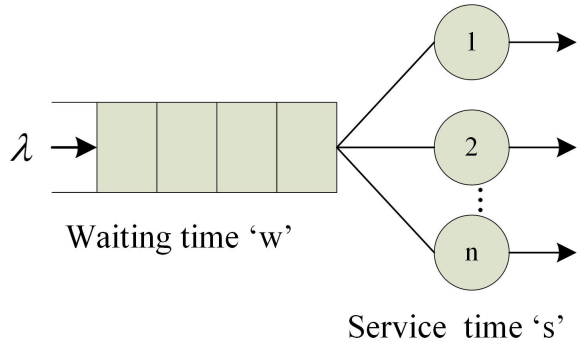

Figure.3 M/M/C Queuing Model

Case 2: When the traffic is continuous. Let the Queuing model adopted be $M|M|C$ as shown in figure 3. where first M represents the arrival rate of requests, second M represents the service rate of requests and C represents the number of controllers. Each controller can serve an infinite number of requests. Let the arrival time of the request R_i and R_{i+1} be t_i andt_{i+1} respectively. The inter arrival time between the packets is represented represented as Δt in equation (20):

$$\Delta t \leftarrow t_{i+1} - t_i \tag{20}$$

The arrival rate of the requests follows erlang distribution which is represented as λ and the service rate is represented as μ.The probability that the new incoming request has to wait for getting service from controller follows erlang distribution represented in equation 21:

$$S(C,p) \leftarrow P_0 * \frac{C^* p^C}{C!(C-p)} \tag{21}$$

$$p = \frac{\lambda}{\mu} \tag{22}$$

$$P_0 \leftarrow \left[\sum_{k=0}^{c-1} \frac{p^k}{k!} + \frac{Ce^c}{C!(C-p)}\right]^{-1} \tag{23}$$

The mathematical model of DDoS attack detection is represented in equation (24):

$$Maximize\ Z = W_1^* Attack\_Detect\_Rate + W_2^* Accuracy \tag{24}$$

Subject to

$$t_{i+1} - t_i > threshold \tag{25}$$

where i represents Request

The attack detection rate is the ratio of the number of attacks detected to the sum of true positives and false positives represented in equation (26):

$$Attack\_Detect\_Rate \leftarrow \frac{TruePositive}{TruePositive + FalsePositive} \tag{26}$$

Accuracy is defined as the ratio of the sum of true positives and true negatives to the total number of instances represented in equation (27):

$$Accuracy \leftarrow \frac{TruePositive + TrueNegative}{\#Requests} \tag{27}$$

Let the number of networking devices be given as |n| under the controller con_j. When the networking device D_i in data plane receives a request, it checks its flow table FT_i if there is an entry in the flow table then the request will be processed, else if there is no entry in the flow table then the request will be forwarded to the controller con_j where it checks its flow table. Assume that the request is from a spoofed IP address, while simultaneously the controller is searching in the flow table, the attacker launches an infinite number of requests. Thus, the entire network is compromised.

In order to protect the control plane, thereby protecting the network, an efficient DDoS detection mechanism is required. From the survey, it is observed that the conventional method of detecting DDoS in SDN has to be enhanced in order to maximize the attack detection rate. Thus, Meta Heuristic clustering based on Whale Optimization algorithm is used to detect the DDoS attack. WOA based clustering bunches the requests as either normal or DDoS attack.

Clustering is an unsupervised learning where the class label of the instance is not known. The conventional clustering has a serious drawback of trapping into local optima which leads to an increase in true negative rate and false negative rate. Thus, Whale Optimization algorithm based clustering is proposed for optimal clustering of normal requests and attacks thereby maximizing the attack detection rate. This, in turn, increases the accuracy.

5 | PERFORMANCE ANALYSIS

In this section, the performance of the proposed WOA-DD algorithm has been analyzed and compared with various existing approaches that detect and prevent the DoS attack. Whenever the request arrives at the request handler, it

checks whether it is from a normal user or from an attacker using WOA-DD deployed in it. If the request is normal, then it is forwarded to the to the data plane which results in maximizing attack detection rate and accuracy. Let the arrival time of the normal user and attacker be given as λ =0.30/s and 1/s respectively. Each request will be serviced at the rate of μ = 0.10/s.

The proposed methodology WOA-DD is implemented in Java and integrated with WEKA. The WOA-DD is compared with naïve Bayes and K-means using WEKA. DDoS attacks are generated and logs are recorded using DDOSIM simulator which is used to create a model for detecting DDoS. Initially, 600 requests are generated using DDOSIM simulator and the corresponding log file after being pre-processed and are given as an input for proposed WOA-DD, K-Means and Naïve Bayes algorithms. WOA-DD outperforms in clustering the fruitful improvements in terms of breach time. When the number of requests is scaled to 1200, the proposed WOA-DD outperforms as well.

A. COMPARISON OF DETECTION OF DDOS

In figure 4, we have compared the algorithms that are used to detect the attack with and without WOA. Figure 4 shows the detection of DDoS in our proposed system. The horizontal axis represents the number of requests and the vertical axis represents the number of requests detected as DDoS attacks. The plot represents the detection of DDoS using Whale Optimization algorithm based clustering.

From figure 4, we can observe that the detection of DDoS in our proposed approach is optimal than the existing approaches and thus it maximizes the attack detection rate. The actual number of attacks from the requests made is retrieved from the data set and the graph reveals the detection of DDoS is efficient using WOA.

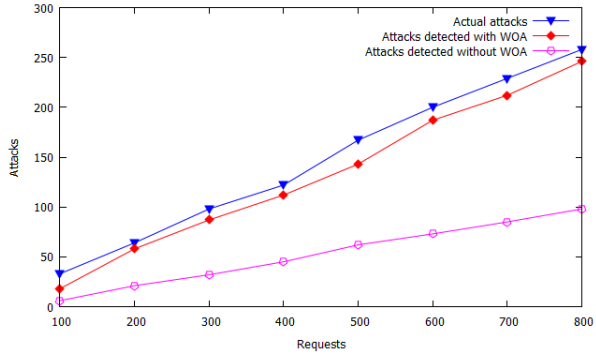

Figure.4 Detection of DDoS

B. COMPARISON OF ACCURACY

Since we are proposing a nature-inspired optimization for attack detection, we compare our proposed algorithm with prevailing machine learning algorithms and the values have been tabulated as Table 1. Table 1 shows the classification accuracy and time required for attack detection in seconds. From the table, it is observed that the WOA-DD achieves better accuracy in minimal time as compared to other supervised Naïve Bayes and Unsupervised K-Means.

The reason behind high optimal detection of DDoS in WOA-DD is that the algorithm iteratively chooses optimal cluster centroid for clustering the attacks from normal requests. Thus, the cluster formed using WOA-DD is optimal which results in the increased detection of attacks, thereby maximizing the accuracy with minimal time.

The Naïve Bayes is a supervised learning approach which deals with the probability. It classifies a request as an attack or normal on the basis of the probability.

The K-Means is an unsupervised learning which aims to cluster the attack and normal request. However, the cluster formed is not optimal as it is biased towards initial cluster centroid. Owing to the above drawbacks, naïve Bayes and K-Means achieve less accuracy and take more time than WOA-DD represented in Table 1 and 2. Table 1 and 2 show the simulation of 600 and 1200 requests respectively.

TABLE 3 Time taken to detect DDoS for the number of requests = 600

Algorithm	Accuracy	Detection Time(S)
Naïve Bayes	0.753333333	1.12
K-Means	0.756666667	1.14
WOA-DD	0.788333333	0.98

TABLE 4 Time taken to detect DDoS for the number of requests = 1200

Algorithm	Accuracy	Detection Time(S)
Naïve Bayes	0.719167	1.42
K-Means	0.720833	1.67
WOA-DD	0.828333	1.28

TABLE 5 Performance of WOA-DD with other machine learning approaches for the number of Requests = 600

Algorithm	TP	FP	TN	FN	Precision	Recall	F-Measure	Accuracy
Naïve Bayes	312	25	140	23	0.925816	0.931343	0.928571429	0.753333333
K-Means	302	17	152	29	0.946708	0.912387	0.929230769	0.756666667
WOA-DD	374	24	99	3	0.939698	0.992042	0.96516129	0.788333333

Even when the number of requests is increased, the proposed WOA-DD outperforms other existing approaches which are represented in Table 3 and 4 for the request of 600 and 1000.

TABLE 6 Performance of WOA-DD with other machine learning approaches for the number of Requests = 1200

Algorithm	TP	FP	TN	FN	Precision	Recall	F-Measure	Accuracy
Naïve Bayes	602	294	261	43	0.671875	0.933333333	0.781310837	0.719166667
K-Means	587	321	278	14	0.646476	0.976705491	0.777998675	0.720833333
WOA-DD	700	166	294	40	0.808314	0.945945946	0.871731009	0.828333333

C. COMPARISON OF SENSITIVITY AND SPECIFICITY

Sensitivity represents the total number of attacks that are correctly identified. Otherwise, they are termed as recall or true positive rate. Specificity represents the ratio of attacks that are correctly identified as attacks. From Table 5, it is observed that the WOA-DD achieves maximum true positive rate and minimum true negative rate. An efficient algorithm should have a maximum true positive rate and a minimum true negative rate.

TABLE 7 Time taken to detect DDoS for the number of requests = 600

Algorithm	True Positive Rate (Sensitivity)	True Negative Rate (Specificity)
Naïve Bayes	0.931343284	0.848484848
K-Means	0.912386707	0.899408284
WOA-DD	0.99204244	0.804878049

D. COMPARISON OF NUMBER OF ENTRIES IN FLOW TABLE

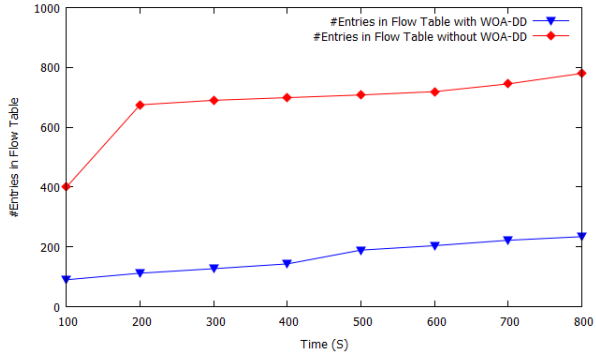

Figure 5. Number of entries in the flow table

In Figure 5, the number of entries in the flow table with respect to time is compared. We infer that the number of entries is decreased drastically in the proposed WOA-DD approach as it is deployed in the request handler whose objective is to check whether it is a normal request or an attack.Thus, the attack is detected before it is forwarded to the data

plane which results in minimal entries in the flow table; consequently, saving cost. Also, it has an impact which ensures attacker is not able to deplete the flow table.

E. COMPARISON OF BANDWIDTH OCCUPANCY

Figure 6 represents the bandwidth occupied between the controller and the switch. In the proposed method, initially the request is detected by WOA-DD as an attack or a normal request. If the request is identified as an attack, it will not be forwarded to the data plane. This, in turn, does not forward the request to the controller which prevents depletion of bandwidth occupied between data plane and control plane.

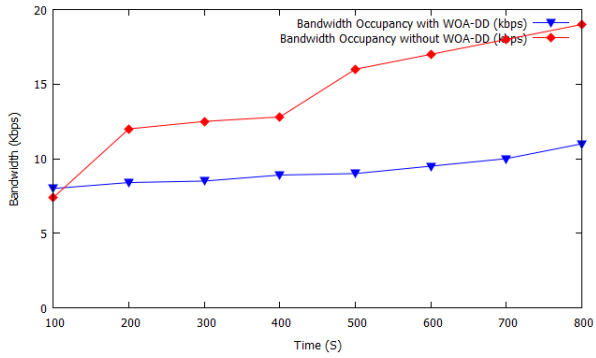

Figure 6. Comparison of Bandwidth Occupancy

F. ANALYSIS OF WOA WITH DIFFERENT BENCHMARK FUNCTION

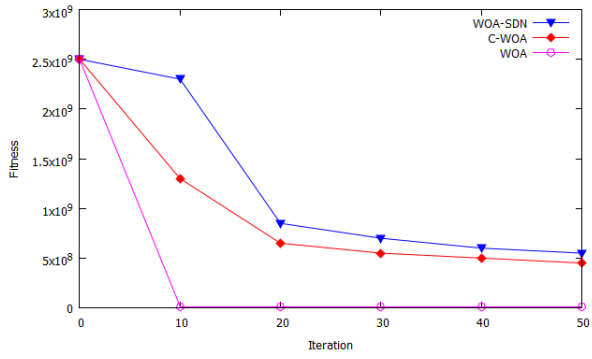

Figure 7. Performance comparison on the F01 Sphere function

Since our proposed WOA-DD algorithm is a nature-inspired Heuristic algorithm, we have compared our algorithm

with various benchmark functions of the same kind. Our analysis shows that adopting WOA-DD is easier than other preexisting schemes. In Figure 7, we have analysed the proposed WOA-SDN algorithm with existing Chaotic – Whale Optimization Algorithm (C-WOA) [27] and WOA [28] algorithm with several benchmark functions by considering fitness value with the number of iterations. We discovered that when WOA is in assistance with the SDN controller, it outperforms. The number of iterations required to compute the fitness function is considerably high which makes this approach more rigid.

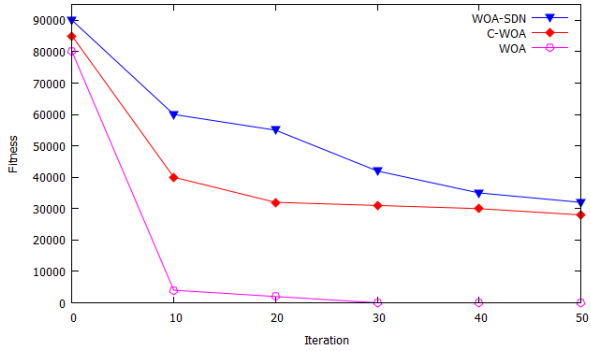

Figure 8. Performance comparison on the Cigar function

In Figure 7 and 8, we have considered Sphere and Cigar function to calculate fitness value based on the variations in the number of iterations. Our proposed WOA-SDN gets an improved fitness value than the existing C-WOA and WOA algorithms. Similarly, in figure 9, we compared the resulting WOA-SDN equipped with our WOA-DD algorithm with the Greiwank function for convergence property of optimization.In figure 10, we have compared our algorithm with a single objective and the multiple objectives for the Schaffer function and we have found a minimum of 60 percent improvements in the fitness values. In figure 11, we have determined the unconstrained global optimization of the learning algorithms with respect to the fitness value for the Schwelfel function. Through various analyses, we conclude that our proposed algorithm outperforms in all the aspects and can be highly scalable for the distributed environment.

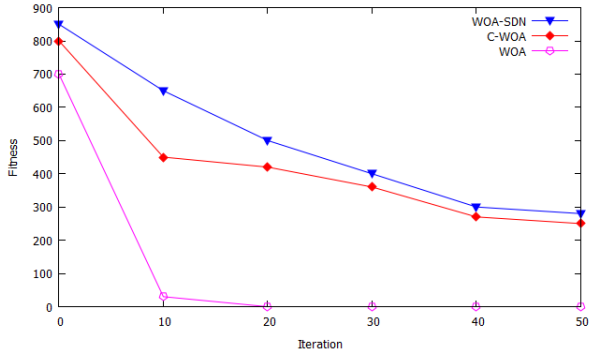

Figure 9. Performance comparison on the Greiwank function

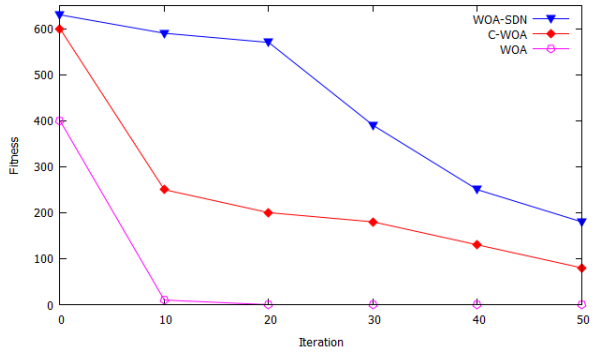

Figure 10. Performance comparison on the Schaffer function

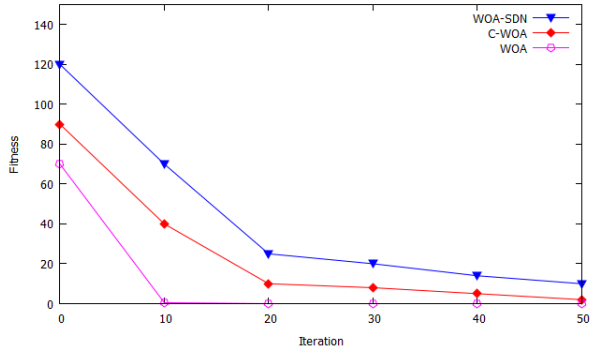

Figure 11. Performance comparison on the Schwefel 2.21 function

6 | CONCLUSION

In this article, we have addressed the key issue that prevails in the SDN assisted networks using a Meta Heuristic approach and have achieved a denial of service free SDN environment. The proposed DoS prevention technique not only avoids the attack but it also paves the way to form a healthy and seamless connectivity through SDN. Though DDoS is inevitable in the networking environment, we have formulated DDoS prevention strategy that effectively avoids DoS attack by an average of 70 percent. Our proposed WOA-DD algorithm is effective in cases where initial cluster formation time can be compromised. Once the clusters are formed, WOA-DD protects the system from most of the attacks like flooding, jamming and DoS. The proposed WOA-DD algorithm has been analyzed for security threats in AVISPA tool and is found to be safe from DoS attacks. Also, we have analyzed the performance of WOA-DD algorithm under various conditions and the results originate to be supporting our claim. We inferred few limitations in this contribution as follows, i) Though WOA-DD avoids DDoS, it incurs a considerable delaying in decision making because of the clustering process, ii) The centroid calculation determines the attack detection, so the fitness function formed should be optimal for better results, iii) The proposed mechanism is best fit in the scenario where network delay is not an issue. The future work of this article is to extend the formed cluster in order to balance the load over the SDN

controllers on a large scale and to improve the security of the SDN controller by adopting blockchain strategies.

ACKNOWLEDGEMENTS

This work was supported by the KAIST GCORE (Global Center for Open Research with Enterprise) grant funded by the Ministry of Science and ICT (Project SDN/NFV and Cloud).

REFERENCES

[1] Karakus M, Durresi A. A survey: Control plane scalability issues and approaches in software-defined networking (SDN). Computer Networks 2017;112:279–293.

[2] Rotsos C, King D, Farshad A, Bird J, Fawcett L, Georgalas N, et al. Network service orchestration standardization: A technology survey. Computer Standards & Interfaces 2017;54:203–215.

[3] Ahmad I, Namal S, Ylianttila M, Gurtov A. Security in software defined networks: A survey. IEEE Communications Surveys & Tutorials 2015;17(4):2317–2346.

[4] Sharma PK, Chen MY, Park JH. A software defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access 2018;6:115–124.

[5] Sharma PK, Singh S, Jeong YS, Park JH. Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks. IEEE Communications Magazine 2017;55(9):78–85.

[6] Bashir AK, Ohsita Y, Murata M. Abstraction Layer based distributed architecture for virtualized data centers. Proc Cloud Computing 2015;p. 46–51.

[7] Lim S, Ha J, Kim H, Kim Y, Yang S. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In: Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on IEEE; 2014. p. 63–68.

[8] Scott-Hayward S, O'Callaghan G, Sezer S. SDN security: A survey. In: Future Networks and Services (SDN4FNS), 2013 IEEE SDN For IEEE; 2013. p. 1–7.

[9] Yan Q, Yu FR. Distributed denial of service attacks in software-defined networking with cloud computing. IEEE Communications Magazine 2015;53(4):52–59.

[10] Ambrosin M, Conti M, De Gaspari F, Devarajan N. Amplified distributed denial of service attack in software defined networking. In: New Technologies, Mobility and Security (NTMS), 2016 8th IFIP International Conference on IEEE; 2016. p. 1–4.

[11] Rafati S, Stiller B. A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts. In: Security of Networks and Services in an All-Connected World: 11th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2017, Zurich, Switzerland, July 10-13, 2017, Proceedings, vol. 10356 Springer; 2017. p. 16.

[12] Bashir AK, Ohsita Y, Murata M. BS-3-9 Orchestration of SDN/NFV Enabled Network Service Chains Using Optoelectronic Routers (BS-3. Advanced Networking Technologies for Innovative Information Networks). 2016;2016(2).

[13] Mousavi SM, St-Hilaire M. Early detection of DDoS attacks against SDN controllers. In: Computing, Networking and Communications (ICNC), 2015 International Conference on IEEE; 2015. p. 77–81.

[14] Kreutz D, Ramos FM, Verissimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: A comprehensive survey. Proceedings of the IEEE 2015;103(1):14–76.

[15] Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software 2016;95:51–67.

[16] Bashir AK, Lim SJ, Hussain CS, Park MS. Energy efficient in-network RFID data filtering scheme in wireless sensor networks. Sensors 2011;11(7):7004–7021.

[17] Wang B, Zheng Y, Lou W, Hou YT. DDoS attack protection in the era of cloud computing and software-defined networking. Computer Networks 2015;81:308–319.

[18] Krishnan R, Krishnaswamy D, Mcdysan D. Behavioral security threat detection strategies for data center switches and routers. In: Distributed Computing Systems Workshops (ICDCSW), 2014 IEEE 34th International Conference on IEEE; 2014. p. 82–87.

[19] Arul R, Raja G, Kottursamy K, Sathiyanarayanan P, Venkatraman S. User path prediction based key caching and authentication mechanism for broadband wireless networks. Wireless Personal Communications 2017;94(4):2645–2664.

[20] Mowla NI, Doh I, Chae K. Multi-defense mechanism against DDoS in SDN based CDNi. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2014 Eighth International Conference on IEEE; 2014. p. 447–451.

[21] Belyaev M, Gaivoronski S. Towards load balancing in SDN-networks during DDoS-attacks. In: Science and Technology Conference (Modern Networking Technologies)(MoNeTeC), 2014 First International IEEE; 2014. p. 1–6.

[22] Dao NN, Park J, Park M, Cho S, et al. A feasible method to combat against DDoS attack in SDN network. In: 2015 International Conference on Information Networking (ICOIN) IEEE; 2015. p. 309–311.

[23] Dharma NG, Muthohar MF, Prayuda JA, Priagung K, Choi D. Time-based DDoS detection and mitigation for SDN controller. In: Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific IEEE; 2015. p. 550–553.

[24] Anbalagan S, Kumar D, Raja G, Ejaz W, Bashir AK, et al. SDN-assisted efficient LTE-WiFi aggregation in next generation IoT networks. Future Generation Computer Systems 2017;.

[25] Yan Q, Yu FR, Gong Q, Li J. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Communications Surveys & Tutorials 2016;18(1):602–622.

[26] Chin T, Mountrouidou X, Li X, Xiong K. An SDN-supported collaborative approach for DDoS flooding detection and containment. In: Military Communications Conference, MILCOM 2015-2015 IEEE IEEE; 2015. p. 659–664.

[27] Dong P, Du X, Zhang H, Xu T. A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In: Communications (ICC), 2016 IEEE International Conference on IEEE; 2016. p. 1–6.

[28] Karthi R, Arumugam S, Kumar KR. Discrete particle swarm optimization algorithm for data clustering. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2008) Springer; 2009.p. 75–88.